<u>**AMENDMENTS TO THE CLAIMS**</u>

This listing of claims will replace all prior versions, and listings, of claims in the application:

<u>**Listing of Claims:**</u>

1   1.   A method for executing a fail instruction to facilitate transactional
2 execution on a processor, comprising:
3   transactionally executing a block of instructions within a program;
4   wherein changes made during the transactional execution are not
5 committed to the architectural state of the processor unless the transactional
6 execution successfully completes; and
7   if the fail instruction is encountered during the transactional execution,
8 terminating the transactional execution without committing results of the
9 transactional execution to the architectural state of the processor, wherein
10 terminating the transactional execution involves branching to a location specified
11 by the fail instruction or to a location specified by a start transactional execution
12 (STE) instruction at the beginning of the transactional execution; or setting state
13 information in the processor and continuing the transactional execution, wherein
14 the processor handles the failure later.
15

1   2.   The method of claim 1, wherein terminating the transactional
2 execution involves discarding changes made during the transactional execution.

1   3.   The method of claim 2, wherein discarding changes made during
2 the transactional execution involves:

3   discarding register file changes made during the transactional execution;

4   clearing load marks from cache lines;

5   draining store buffer entries generated during transactional execution; and

6   clearing store marks from cache lines.

1   4-5.

1   6. The method of claim 1, wherein terminating the transactional

2 execution additionally involves attempting to re-execute the block of instructions.

1   7. The method of claim 1, wherein if the transactional execution of

2 the block of instructions is successfully completed, the method further comprises:

3   atomically committing changes made during the transactional execution;

4 and

5   resuming normal non-transactional execution.

1   8. The method of claim 1, wherein potentially interfering data

2 accesses from other processes are allowed to proceed during the transactional

3 execution of the block of instructions.

1   9. The method of claim 1, wherein if an interfering data access from

2 another process is encountered during the transactional execution, the method

3 further comprises:

4   discarding changes made during the transactional execution; and

5   attempting to re-execute the block of instructions.

1   10. The method of claim 1, wherein the block of instructions to be

2 executed transactionally comprises a critical section.

1   11.   The method of claim 1, wherein the fail instruction is a native

2   machine code instruction of the processor.

1   12.   The method of claim 1, wherein the fail instruction is defined in a

2   platform-independent programming language.

1   13.   A computer system that supports a fail instruction to facilitate

2   transactional execution, comprising:

3   a processor; and

4   an execution mechanism within the processor;

5   wherein the execution mechanism is configured to transactionally execute

6   a block of instructions within a program;

7   wherein changes made during the transactional execution are not

8   committed to the architectural state of the processor unless the transactional

9   execution successfully completes; and

10   wherein if the fail instruction is encountered during the transactional

11   execution, the execution mechanism is configured to:

12   terminate the transactional execution without committing results of

13   the transactional execution to the architectural state of the processor,

14   wherein terminating the transactional execution involves branching to a

15   location specified by the fail instruction or to a location specified by a start

16   transactional execution (STE) instruction at the beginning of the

17   transactional execution, or to set state information in the processor and

18   continue transactional execution, wherein the execution mechanism is

19   configured to handle the failure later

1   14.   The computer system of claim 13, wherein while terminating the

2 transactional execution, the execution mechanism is configured to discard changes

3 made during the transactional execution.

1   15.   The computer system of claim 14, wherein while discarding

2 changes made during the transactional execution, the execution mechanism is

3 configured to:

4   discard register file changes made during the transactional execution;

5   clear load marks from cache lines;

6   drain store buffer entries generated during transactional execution; and

7   clear store marks from cache lines.

1   16-17.

1   18.   The computer system of claim 13, wherein while terminating the

2 transactional execution, the execution mechanism is additionally configured to

3 attempt to re-execute the block of instructions.

1   19.   The computer system of claim 13, wherein if the transactional

2 execution of the block of instructions is successfully completed, the execution

3 mechanism is configured to:

4   atomically commit changes made during the transactional execution; and

5 to

6   resume normal non-transactional execution.

1   20.   The computer system of claim 13, wherein the computer system is

2 configured to allow potentially interfering data accesses from other processes to

3 proceed during the transactional execution of the block of instructions.

1   21.   The computer system of claim 13, wherein if an interfering data

2 access from another process is encountered during the transactional execution, the

3 execution mechanism is configured to:

4   discard changes made during the transactional execution; and

5   attempt to re-execute the block of instructions.

1   22.   The computer system of claim 13, wherein the block of

2 instructions to be executed transactionally comprises a critical section.

1   23.   The computer system of claim 13, wherein the fail instruction is a

2 native machine code instruction of the processor.

1   24.   The computer system of claim 13, wherein the fail instruction is

2 defined in a platform-independent programming language.

1   25.   A computing means that supports a fail instruction to facilitate

2 transactional execution, comprising:

3   a processing means; and

4   an execution means within the processing means;

5   wherein the execution means is configured to transactionally execute a

6 block of instructions within a program;

7   wherein changes made during the transactional execution are not

8 committed to the architectural state of the processor unless the transactional

9 execution successfully completes; and

10   wherein if the fail instruction is encountered during the transactional

11 execution, the execution means is configured to:

12   terminate the transactional execution without committing results of

13   the transactional execution to the architectural state of the processor,

6

14   wherein terminating the transactional execution involves branching to a

15   location specified by the fail instruction or to a location specified by a start

16   transactional execution (STE) instruction at the beginning of the

17   transactional execution, or to set state information in the processor and

18   continue transactional execution, wherein the execution means is

19   configured to handle the failure later.

20